



# Painting and Non-Photorealistic Graphics

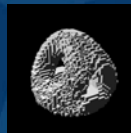
## Painting and Rendering Textures on Unparameterized Models

David DeBry  
Jonathan Gibbs  
Devorah DeLeon Petty  
Nate Robins



### Overview

- Why?
- The Structure of the Texture
- The User's View
- Implementation Details
- Results
- Current Projects / Future Work



### Related Work

#### 3D Paint Systems

Vertex Colors: Hanrahan & Haeblerli 1990

Texture Maps: Daily & Kiss 1995

#### 3D Texture Mapping

Peachy 1985

Perlin 1985

#### Automatic UV Assignment

Levy et al. 2002

#### Hierarchical Images & Volumes

Octrees: Samet 1990

Multiresolution 2D painting: Berman et al. 1994

Volume Rendering: Wilhelms & Van Gelder 1994

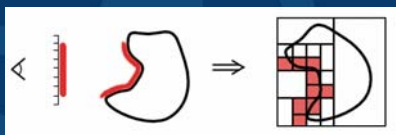


### Idealistic Targets

- No explicit parameterization
- Minimize seams and distortions
- Minimize modeling/texturing iterations
- Adaptive detail
- Memory use similar to 2D
- Minimize changes for pipeline and artists



### Octex: Texture map as an octree



#### Basic Construction

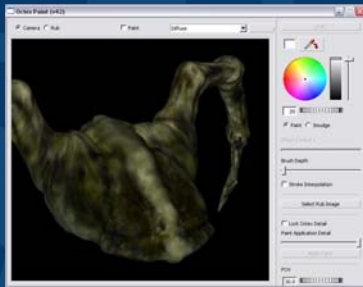
- Contains multiple maps
- Node may contain more than one color sample for a given map
- Pointerless octree [Samet 90]



### 3D Paint Tool - Interaction

No hardware support for octtextures,  
so how to paint?

### 3D Paint Tool - Interaction



#### Model motion: Hardware

Vertex colors sampled from Octex

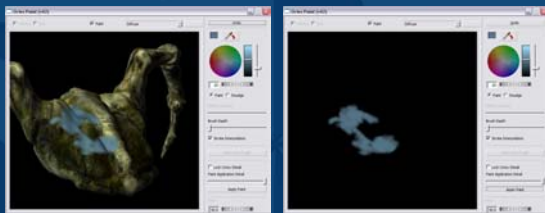
### 3D Paint Tool - Interaction



#### High-quality beauty renders: Software

Internal software render

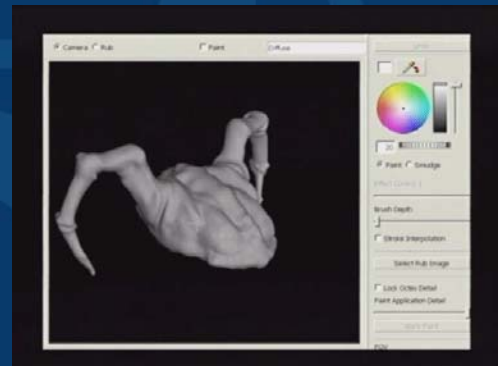
### 3D Paint Tool - Interaction



#### Brush motion: Hybrid

Various HW and SW screen space buffers

### 3D Paint Tool



### Octex Growth



Don't make the user fight the software

- Hard limits are constraining
- Tree depth limits are non-intuitive

### Octex Growth



Artist-intuitive controls

- Detail level: pixel-to-node ratio
- Detail lock: no new nodes

## Paint Insertion - Problems



- **Scan convert paint**

Screen space sampling results in aliasing in the octex

- **Iterate over geometry**

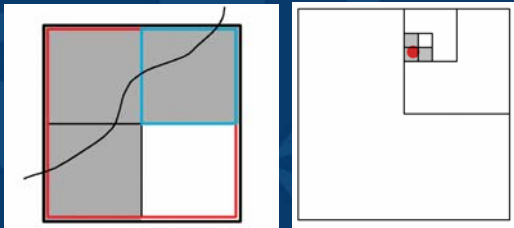
Abutting geometry causes double insertion

## Paint Insertion - Solution



- **Bin the painted geometry into nodes resolving per-pixel occlusions, subdividing nodes as needed**
- **Per node, composite screen space paint over existing node color – utilize both transparency and coverage**

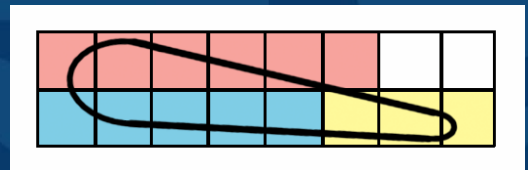
## Paint Insertion



### Creating child nodes

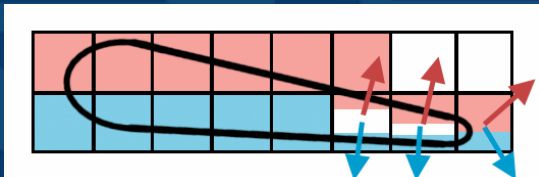
- **Potential children based on geometry**
- **Existing children based on paint**

## Paint Leaks



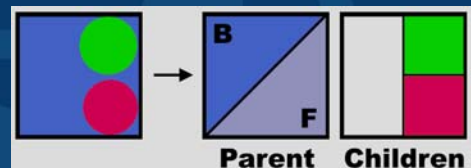
Very thin objects cause octex to increase detail in order to prevent paint leaks

## Paint Leaks



Optionally include additional color samples with associated quantized normal

## Sampling and Filtering



### Parent's Basic color:

Color of *uncreated* children

### Parent's Filtered color:

Average of *created* and *uncreated* children

3d equivalent of a MIP-map

## Memory



Memory use varies from 2D texture maps by only a constant factor

- Compare number of nodes to number of pixels
- How much of the 2D texture is oversampled?

All comparisons depend on quality of UVs

## Memory



Ideal UV mapping:

- No unused space
- Distribution choices:
  - All the pixels are the same size in object space (even detail)
  - Density of pixels matches density of painted detail

## Memory



Extreme Case:

The leaves of the nodes match the pixels in the flat map, but we have the overhead of the tree.

This gives us 1/3 more nodes than pixels (with typical geometry)

## Memory



Typical Case: "The 1/3 Rule"

Comparing the Octex to the typical UV mapping where equal area is the goal, if only 1/3 of the model ends up one level less detailed in the tree, the Octex has fewer nodes than the flat map has pixels.

## Memory



Clean-up passes:

- Remove children matching parent
- Parents without children lose second color
- Remove children similar to parent (lossy compression)

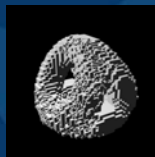
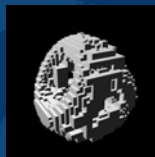
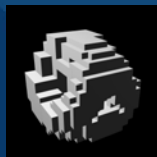
## Results – Our First Image



Results - Our First Good Image



Results - Increasing detail



Results - 2D Textures



Results - Octex



Results - Octex



Results - Octex



## Results – Octex



## Results - Octex



## Other Projects



ESC Entertainment  
“Matrix” sequels

Open Source  
Core library – MIT License  
Tools – GPL License  
[thrownclear.com](http://thrownclear.com)

## Future Work



- Handling changes in model
- Improved filtering and sampling
- Don’t confine paint to the surface
- Hardware implementation – shaders?

## Acknowledgements



Mark Edwards – PDI/Dreamworks  
Scott Peterson – PDI/Dreamworks

FOX GUI – [fox-toolkit.org](http://fox-toolkit.org)

## Painting and Non-Photorealistic Graphics



### Painting and Rendering Textures on Unparameterized Models

David DeBry  
Jonathan Gibbs  
Devorah DeLeon Petty  
Nate Robins